DECISION SUPPORT SYSTEMS
AND
PERSONAL COMPUTING

Peter G. W. Keen
Richard D. Hackathorn

October 1979

CISR No. 47
Sloan WP No. 1088-79

Center for Information Systems Research
Massachusetts Institute of Technology
Alfred P. Sloan School of Management
50 Memorial Drive
Cambridge, Massachusetts, 02139
617 253-1000

DECISION SUPPORT SYSTEMS
AND
PERSONAL COMPUTING

Peter G. W. Keen
Richard D. Hackathorn

October 1979

CISR No. 47
Sloan WP No. 1088-79

ABSTRACT

This paper explores the relationship between Decision Support Systems and Personal Computing. The aim is to clarify the key issues relevant to helping managers make effective use of computer technology in their own jobs. The paper suggests that there exists a useful distinction between Personal Support (PS) and Organizational Support (OS). This distinction is critical for designing a system for effective decision support. The emphasis is placed on Personal Support by discussing development and research issues. Finally, it is suggested that Personal Computing lacks a conceptual base for exploiting the potential of small-scale computer technology and that Personal Support can provide such a framework.

OUTLINE

1.0 MOTIVATION FOR DECISION SUPPORT
- 1.1 DECISION SUPPORT SYSTEMS
- 1.2 PERSONAL COMPUTING
- 1.3 CONCLUSIONS FROM PREVIOUS RESEARCH

2.0 DEFINITION OF SUPPORT SYSTEMS
- 2.1 UTILIZING COMPUTER TECHNOLOGY
- 2.2 LEVELS OF SUPPORT SYSTEMS

3.0 FOCUS ON PERSONAL SUPPORT
- 3.1 COMPONENTS OF A SYSTEM FOR PERSONAL SUPPORT
- 3.2 BUILDING, USING, AND EVOLVING
- 3.3 THE DYNAMICS OF INTERACTION

4.0 RESEARCH ISSUES RELATED TO PERSONAL SUPPORT
- 4.1 IDENTIFYING IMPORTANT VERBS
- 4.2 LACK OF GOOD RESEARCH METHODOLOGY

5.0 CONCLUSIONS

ACKNOWLEGEMENT

The authors wish to acknowledge the valuable discussions with faculty and students of the BA508A graduate seminar on "Personal Support Systems" during the spring of 1979 at the Wharton School, University of Pennsylvania.

1.0 MOTIVATION FOR DECISION SUPPORT

Decision Support Systems (DSS) are interactive computer aids designed to assist managers in complex tasks requiring human judgment. The aim of such systems is to support and possibly to improve a decision process. Personal Computing (PC) is a more recent term that describes the direct use of small-scale systems by an individual for any information processing task. The user has complete control over all aspects of the technology: access, usage, program development and data management.

This paper explores the relationship between DSS and PC. The aim is to clarify key issues relevant to helping managers make effective use of computer technology in their own jobs. The paper provides an overview of DSS and suggests a distinction between Personal Support (PS) and Organizational Support (OS) that in itself seems critical to effective application of the concept of Decision Support. The paper then discusses design and development of systems for personal support and concludes with an assessment of key research issues. The overall objective is: (1) to define where Personal Computing fits into Decision Support; and (2) to exploit the opportunities that Personal Computing provides

1.1 DECISION SUPPORT SYSTEMS

The DSS movement began about ten years ago. It was largely stimulated by MIT's Project MAC, which both provided access to computing power to the individual user for the first time and pointed towards a human-machine symbiosis that promised vast increases in our ability to handle complex problems. [Licklider, 1965] There have been several studies of DSS usage in organizations. [Alter, 1979; Keen and Scott Morton, 1978] A coherent conceptual framework for Decision Support is emerging, and practitioners are extending the academic work of DSS by building innovative systems to support a range of managerial tasks.

Decision Support requires a detailed understanding of the manager's habits, needs, and concepts. Unlike OR/MS and much of traditional MIS, Decision Support is based on descriptive paradigms of decision making rather than prescriptive and rationalistic perspectives. A central theme in Decision Support is that one cannot improve something one does not understand. The act of "supporting" a manager implies a meshing of analytic tools into his or her existing activities.

The term "Decision Support Systems" is partly a rallying cry. There is no formal theory of Decision Support as yet; theory largely emerges from practice in this applied field. Now that many systems have been built and a number of conceptual studies completed, we can begin to generalize from experience and thus adequately define the field. It is becoming apparent that the key word is "support" and that the term "decision" may be context-free, misleading, or even inaccurate.

Most of the conceptual literature on DSS has emphasized the individual manager. Its theoretical base comes from cognitive psychology; the focus is on human problem-solving, rather than organizational decision making. It is clear, however, that practitioners are far more concerned with organizational issues. The systems that they need are built around activities that involve planning and coordination. While the priesthood of the DSS faith prescribes a system to support a manager's budget <u>decision</u>, the lay missionaries build a system to support the organization's budget <u>process</u>.

1.2 PERSONAL COMPUTING

The term "Personal Computing" is also a new rallying cry. Born out of the hectic enthusiasm of computer hobbyists, the PC area has emerged from minor applications of microprocessors to having broad impacts on many segments of society. [Isaacson et.al., 1978] Admittedly pushed along by new technology, PC is taking on more substantive meaning and brings new insights to the fundamental issue of linking computing technology and the individual. Hackathorn [1978] defines Personal Computing as an information processing activity in which the end user has direct personal control over all stages of the activity. This definition implies a personal relationship to the technology that complements the Personal Support side of DSS. PC shares many of the same aims as DSS, while emphasizing small-scale technology and localized systems. The experience and ambitions of those who march under the PC banner have much to offer true believers in DSS. The hardware and software tools developed by PC can similarly extend the applications of

DSS, and decision support provides an experience base and design criteria for PC.

1.3 CONCLUSIONS FROM PREVIOUS RESEARCH

The Personal Computing field is very new and lacks research to build upon [Niles, 1978]. However, there have been several detailed descriptions of DSS usage, and sufficient experience has been gained to permit at least the following conclusions:

1) 'Easy' applications of DSS (i.e., ones in which the likelihood of success is high) are those which involve such functional areas as finance, where it is simple to find aspects of the problem- solving process that can be better handled by the machine than by the manager. In general, it is most practicable to build a DSS around a micro decision (e.g., the selection of a product price or setting an advertising budget) or convenient retrieval of data (e.g., an automated file drawer).

2) 'Hard' applications by contrast are those where there are many interdependencies, as in strategic planning, and where a manager's problem solving or decision is only part of a more complex process. Macro decisions, such as creating a divisional plan with interdepartmental activities and multicriteria problems, pose substantial difficulties in implementation. It should be noted that easy applications generally involve PS and the hard ones involve OS. Practitioners want to tackle the hard problems and may not gain much insight from experience with PS.

3) DSS evolve. The 'final' system is very different from the original one. New tools shape new uses and vice versa. Managers learn from a DSS and often evolve imaginative new applications. This means that the initial design is partly an experiment; a good DSS contains mechanisms for its own obsolescence (or evolution depending on one's perspective).

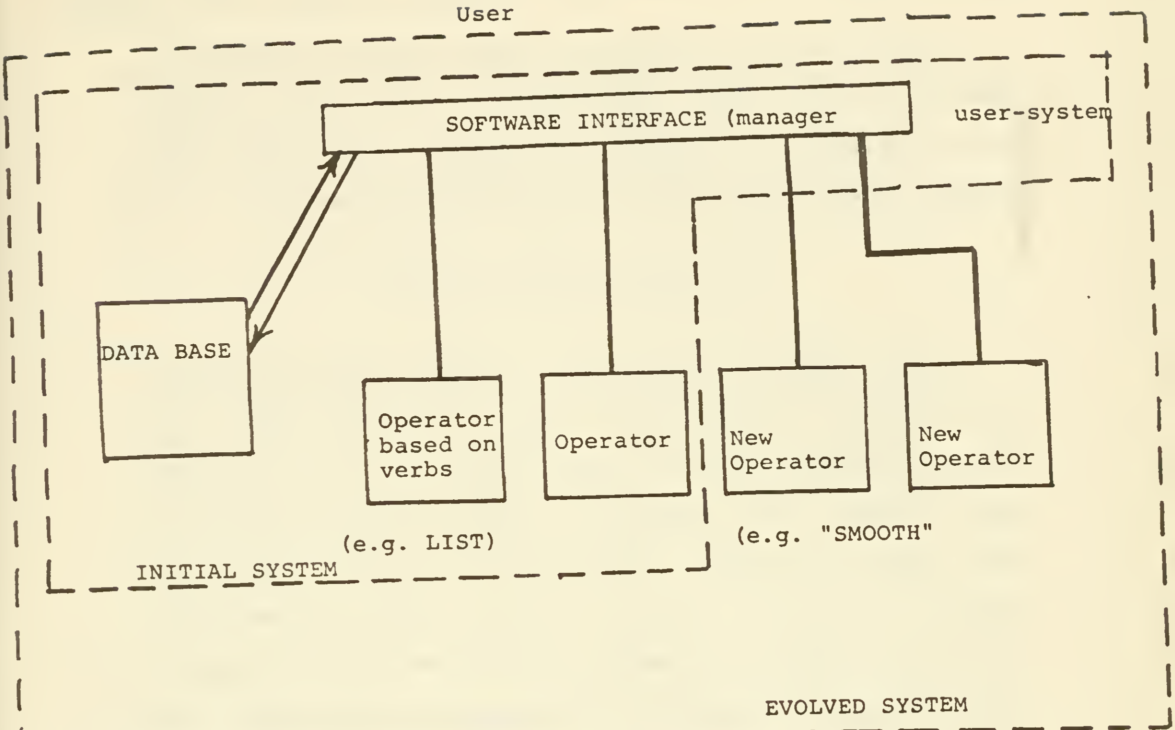
4) The evolutionary nature of DSS has led to a design strategy that emphasizes getting started through a flexible system that can be easily and quickly changed and extended. A common structure has emerged: a DSS consists of a software interface that manages the user-system dialog and a set of modular routines that correspond to verbs or operators (e.g., "display", "compare", "extrapolate", "find", or "smooth"). Evolving a DSS largely means adding new operators, as shown in Figure 1.

[Insert Figure 1 here]

5) DSS are hard to evaluate since they provide mainly qualitative benefits and take a long time to institutionalize. Furthermore, 'improved' decision making is not necessarily the same as better decision outcomes. A DSS can not be simply plugged into the organization but requires a complex process of adjustment and adaptation.

6) Support involves a very detailed understanding of the decision process, task, user, and organizational setting in which the system is to be embedded.

Figure 1 DSS Structure



	Dimensions of Quality
SOFTWARE INTERFACE:	Ease of use Flexibility Helpfulness
OPERATORS	Analytic power Compatibility with User's problem-solving process (correspondence with user's verbs)

It seems clear that a standardized design structure has emerged from independent efforts to build DSS and that it is the notion of support and its corollaries -- evolution and learning -- that makes a DSS different from interactive models or online information systems.

2.0 DEFINITION OF SUPPORT SYSTEMS

One major problem in defining Support Systems is that this area has been principally characterized by computer techology; however, the concept of Decision Support does not necessarily imply a computer-based system. It may be more meaningful to emphasize Decision Support and then to evade the issue by stating that a DSS should be based on any available and suitable technology. Decision Support involves finding criteria for defining ´suitable´. On the other hand, ´available´ technology has meant time-sharing or minicomputers until recently. The technology is now changing very rapidly to include computer networks and microcomputers. The basic objective of applying the technology to support decision processes remains the same.

2.1 UTILIZING COMPUTER TECHNOLOGY

The concept of support has been contrasted with that of ´replacement´ in the DSS literature [Keen & Scott Morton, 1978]. Models, graphics, analytic techniques, software, and hardware are all potential tool that need to be matched to their context. Simply building a system in no way guarantees its use. For the manager, there are ways of applying quantitative methods and computer technology:

1) Ignore them and do nothing. There have been so many examples of insensitive, unrealistic, and incorrect application of technology to management tasks that it is arguable that in many situations the managers' own experience and judgment is preferable and more effective than any computer system [Hoos, 1972].

2) Replace decision making. This is the traditional aim of Operations Research, Management Science, and data processing. The technology is used to rationalize, automate, and establish rules and procedures for particular tasks. Thus, a credit scoring model can replace the loan officer's analysis and judgment (although usually the officer can override the model).

3) Support the decision process. This clearly falls between the extremes of doing nothing and replacing human involvement. Support means augmentation of existing processes, improvement, education, and provision of ad hoc tools that can be integrated into problem solving, communication, and analysis.

A key difference between replacement and support, that in itself implies a difference design approach, is that the former aims at solving a problem or getting an answer while the latter focuses on helping a person. For replacement the area of emphasis is the task or decision itself. A complete system is needed that allows the whole task to be performed through the system. There can be no concept of learning and adaptation, since the design is centered around the task structure. By contrast, support suggests ongoing development so that the main aim is to begin and facilitate the process of augmentation and improvement.

The 'best' system for replacement is one that improves efficiency and/or generates a better solution. For support, a 'good' system improves procedures and processes. It may be hard to link these to outcomes and to 'hard' benefits. Because of this, there can rarely be a reliable prediction of system uses and hence of costs and benefits. Obviously, a proposal to support a task must be justified in much the same way as one to replace decision making but in this situation, the estimates of costs and benefits and outlines for design are essentially a Research and Development activity. It is foolish to plan this activity as if it were a predictable and final venture. Instead, one uses a phased strategy and assumes that some preliminary experiment is needed that can be carefully evaluated and that provides direct guidelines for further development. In general, this initial phase will be small-scale and based on prototypes. The replacement strategy is one of creating products; these may be complex and innovative but are not exploratory and evolutionary. The R & D nature of Decision Support implies distinctive techniques for system building:

1) the main problem is to get started, to build a complete 'first' system that can then be evolved
2) There must be continual learning, in a literal sense. For PS the area of study is the users' problem-solving processes; and for OS information flows and needs for coordination.

2.2 LEVELS OF SUPPORT SYSTEMS

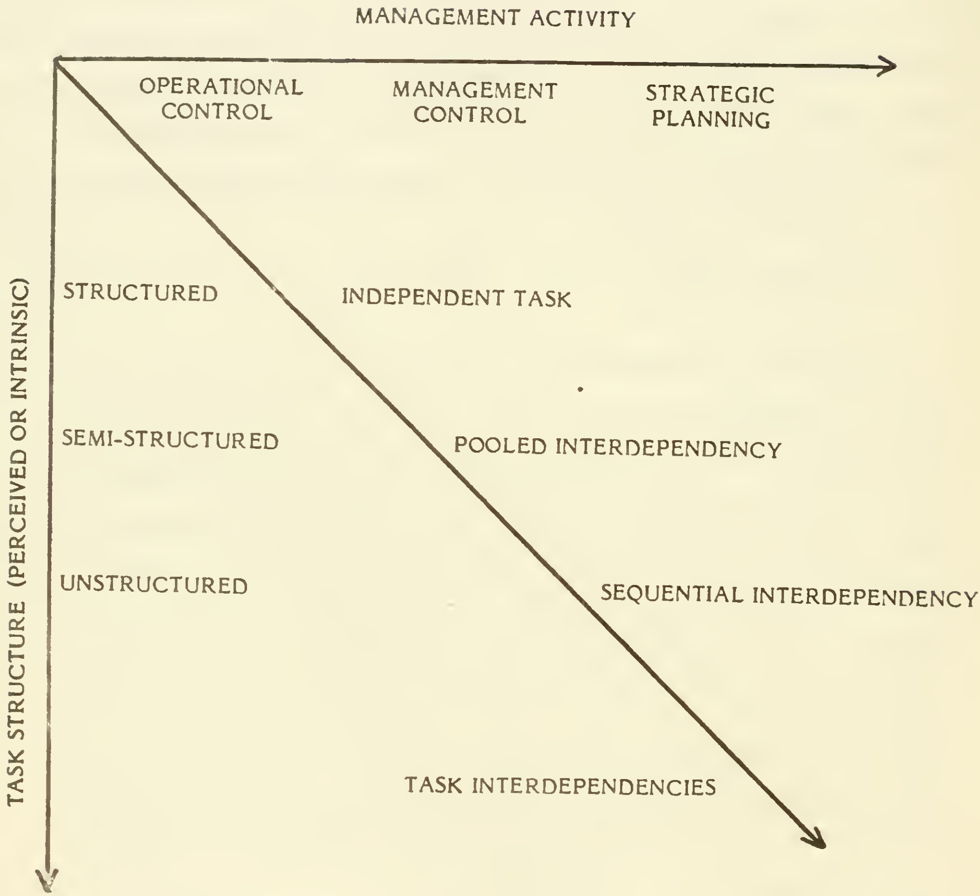
A frequently cited framework for discussing DSS has been Gorry and Scott Morton's [1971], which combines Simon's dichotomy of task structuredness (i.e., structured versus unstructured) with Anthony's classification of managerial activities (i.e., operational control, management control, and strategic planning). This framework seems to apply equally well to both extremes of PS and OS. The main proposal of this paper is that a third dimension -- task interdependency -- should be added, as shown in Figure 2.

[Insert Figure 2 here]

The most obvious characteristic of situations in which DSS has been successful implemented to assist managers (i.e., the 'easy' applications discussed in Section 1.3) is that the user can make a complete decision with the aid of the system. There are no interdependencies with other actors or units. By contrast, several DSS recently developed to support planning activities in large organizations involve tasks that are highly dependent on other organizational units. This type of interdependency is called "sequential" by Thompson [1967] and involves a sequence of decisions, each of which can not be carried out until the preceding one passes on some output. Between sequential and independent lies "pooled" interdependency, which is more complex and fluid. Activity A requires inputs from Activity B but also passes data back to A. A process with pooled dependency is highly interactive among the activities.

Examples of sequential, pooled, and independent tasks in relation to the Anthony framework are shown in Figure 3.

Figure 2


MANAGEMENT ACTIVITY
OPERATIONAL CONTROL
MANAGEMENT CONTROL
STRATEGIC PLANNING
TASK STRUCTURE (PERCEIVED OR INTRINSIC)
STRUCTURED
SEMI-STRUCTURED
UNSTRUCTURED
INDEPENDENT TASK
POOLED INTERDEPENDENCY
SEQUENTIAL INTERDEPENDENCY
TASK INTERDEPENDENCIES

[Insert Figure 3 here]

Where the task does not involve any interdependencies, the manager can reach a decision through his or her own private analysis, though when the choice has been made, it must obviously be communicated to others. The point is that the problem-solving activities do not involve interdependencies. In this situation, Personal Support may be of great value and the quality of the decision improved by the provision of a system designed for direct and individual use by the manager.

Where the task involves sequential interdependencies, each individual's activities must mesh closely with others'. Any computer-based aid will be as much as a vehicle for communication and coordination as for problem-solving. For example, Organizational Support for budgeting will need to provide facilities for storing disaggregated, detailed budgets at the department level, allow these to be integrated first by division and then for the company as a whole.

Pooled interdependency requires face-to-face negotiations and discussions. No single unit can carry out its work without interaction with others. This suggests an additional type of support -- Group Support (GS). This allows users to develop joint plans through a system that provides fast response and that is designed to permit easy exploration of alternatives and explanation of analysis.

The three types of support -- Personal (PS), Group (GS), and Organizational (OS) -- all share a common aim. However, it is essential to distinguish among these types of support as distinct, but related,

FIGURE 3. Expanded Gorry-Scott Morton Framework

		Independent	Pooled	Sequential
perational ontrol	Structured	Order filling	Scheduling room assignments	order entry
	Unstructured	Notice writing	Scheduling management meetings	Shipping and receiving
nagement ntrol	Structured	Task scheduling	Task assignment	Project Costing
	Unstructured	Promotion decision	Personnel matters	Budget formulatio
rategic anning	Structured	Office Arrangement	Factory location	Factory building
	Unstructured	Time Scheduling	Top Management hiring	Merger

components of a Support System:

1) PS focuses on a user or class of users in a discrete task or decision (e.g., setting a price, selecting a stock) that is relatively independent of other tasks.
2) GS focuses on a group of individuals , each of whom are engaged in separate, but highly interrelated, tasks (e.g., office activities).
3) OS focuses on an organizational task or activity involving a sequence of operations and actors (e.g., building a divisional marketing plan, capital budgeting).

A PS may be used within an OS. For example, each manager may use a small-scale system to help set his own marketing budget, and the divisional staff then coordinates and integrates these budgets using an OS. In both cases the aim of the system is to support the user's decision making; the key difference is that PS focuses on a person, while OS is on an organizational process.

While the conceptual literature on DSS emphasizes Personal Support, several of the most successful applications of Decision Support involve Group Support and, more recently, Organizational Support. For example, the DSS built by Scott Morton [1971] which facilitated marketing and production planning provides Group Support. It allowed managers from two departments to come together and use the system as a means of making tradeoffs, sharing ideas, and reaching a consensus.

There are at least a few DSS, designed for Personal Support, that have been difficult to institutionalize because the broader task really involved sequential interdependency. For example, BRANDAID is a system

that in one company substantially facilitated the development of marketing plans by individual brand managers [Keen & Scott Morton, 1978]. Their plans were integrated at the next level of the organization. This required a very different process of analysis. In particular, the problem of 'cannibalization' (i.e, the plan for an individual brand might result in sales being gained at the expense of another of the company's products) had to be addressed. In addition, the managers responsible for integrating the plans required brand decisions to be presented in a format that was not directly compatible with the outputs from BRANDAID. This whole process needed Organizational Support in addition to or in place of this Personal Support.

This concept of levels of support seems important, not only in relation to DSS. For instance, Office Automation (OA) is a packaging of some differentiated technical building blocks very much like DSS. The conceptual literature on OA emphasizes Personal Support, such as text-editing, document preparation, and electronic mail. These are tools that assist and appeal to professional 'knowledge workers.' Practitioners are far more concerned with Organizational Support. They wish to use OA to streamline order entry operations, coordinate decentralized activities, and improve the efficiency of large-scale administrative functions. The literature on OA does not seem to acknowledge the differences in priority and focus; hence, differences in the design criteria for Personal and Organizational Support are also not acknowledged.

3.0 FOCUS ON PERSONAL SUPPORT

The discussion above provides a more differentiated definition of support systems and clarifies where PC and DSS can be meshed. The rest of this paper focusses on Personal Support, which has been only peripherally dealt with in the DSS literature. The empirical studies published so far provide very little insight into PS. Of the six systems described by Keen and Scott Morton[1978], only one, PROJECTOR, was really used for Personal Support. On the whole, only large organizations have been able and willing to undertake the risky effort of implementing experimental systems. In addition, the emergence of small-scale microcomputers, which has created the Personal Computing field, is very recent. We now have a technology for PS but need to shape the concepts necessary for building them. Decision Support provides some guidelines but few examples.

3.1 COMPONENTS OF A SYSTEM FOR PERSONAL SUPPORT

A system for Personal Support can be viewed as analogous to a staff assistant to whom a manager issues commands -- "Do this!". [Keen, 1976] A major research issue is how to identify the user's verbs and build an interface that provides the same quality, flexibility, and ease of communication expected with a staff assistant. For OS, the research questions are more complex, and there has been less empirical or conceptual work to build upon. How can a range of users be supported? Should the interface be more structured and standardized? How can a DSS be meshed into the organizational context of communications, control

systems, and demand for coordination?

From the perspective of a staff assistant, the components for Personal Support divide into three main parts:

1) The Interface: Is the manager able to talk with the PS as with a staff assistant? How much does he or she have to learn to structure the dialogue?

2) The Operators: Do they correspond to the managers' command: "do this". Obviously, the comparison is not a literal one, but it provides some useful criterial for design. For example, if the interface does not facilitate a simple dialogue in which the PS can explain its responses if necessary (through a "help" command) or is not built around the manager's vocabulary, it cannot support him or her in the way an assistant does. Similarly, if the operator cannot be directly related to a verb, it is unlikely that it will be used since: (1) it cannot easily be integrated into the user's concepts and activities; and (2) it is not clear what it supports. For Organizational Support, the corresponding analogy is with a staff coordinator. In both cases, the distinguishing feature of the human is flexibility and adaptivity. The quality and usefulness of the support system similarly depends on these attributes.

3) The Database: The data base is equivalent to the assistant's file cabinet or a library. Clearly, the contents, indexing, and access method determines the range, flexibility, cost, and adaptability of uses.

3.2 BUILDING, USING, AND EVOLVING

Personal support implies no "final" or predefined system. Whereas one may be able to identify the necessary steps and modes of use for an organizational process (i.e. in building OS), it seems clear that the necessary focus is on the person when developing a PS. This implies the need to:

1) design a dialog first;
2) provide a preliminary set of operators;
3) permit the manager to learn from the system; and
4) to evolve the PS by extending its operators.

Learning and flexibility seem to be central aspects of support. They suggest a design sequence similar to Ness´s definition of "middle-out". On the basis of observation of the uses and discussions which aim at identifying key verbs, the designer builds an interface, which is complete enough to permit a meaningful user-system dialogue but which will certainly need later alteration or extension. A preliminary set of operators is implemented. The user now has a concrete system with which to react, and the designer has a vehicle by which to verify his design.

The natural sequence of development of a system for PS is:

1) Using existing operators: The initial system is a complete one, not a prototype. The operators reflect the verbs that the user relied on prior to the development of the PS or ones that are easy to use but that also provide new capabilities.

2) Constructing sequences of operators: If the tool is of any value, it is likely the user will soon develop sequences of analysis, will in effect build up higher-level routines (e.g.: in analyzing a customer account: first use "STATUS" then "REPORT", "PROFIT"; or when reviewing monthly results: use "TREND", "SUMMARIZE", "PLOT"). These sequences will tend to be user-specific.

3) Developing new operators: A key assumption here is that as the user gains experience and confidence in the system, he or she will be ready for (or even specify) new operators. Learning leads to new verbs.

4) Changing existing operators: Fine-tuning a PS may mean modifying operators, or occur as a result of the user's learning (e.g., adverbs qualify a verb and introduce differentiated subfunctions within existing operators.

5) Adding data about objects: In general, items in the database are nouns. The phrase "graph profits versus sales expenses" translates into an operator "GRAPH" with the two nouns "PROFIT" and "SALES EXPENSES" being its arguments.

6) Changing data structures relating to objects: The database is a retrievable set of objects which may need a particular organization or set of definitions for the interface to identify them and associate them with the operators that act on them.

These six aspects of use and evolution provide a link between the behavioral concept of support and the technical realization of that concept, a support system. Support begins by identifying verbs and nouns. Learning, and hence evolution of the Support System, similarly

involves new verbs. An operator must be defined in terms of a verb and vice versa for the system to evolve.

It needs to be acknowledged that the first four items on the above list are easy to handle with existing software. An interface can be redesigned without affecting the operators; the system seems entirely different to the user even though it performs the same functions. Operators can be added and altered, especially if a flexible development language, like APL, is used. However, it is far more difficult to manage and change data structures. There is little discussion in the DSS literature on database design and the tools for data storage in PC are at present very limited.

3.3 THE DYNAMICS OF INTERACTION

The discussion above implies both interactive use of a PS and interactive development. The design process relies on evolution. Clearly, the key first step is to identify "good" verbs. Berry [1977], whose discussion of "levels of language" in the use of APL parallels that in section 3.2 above, gives the example of an economist whose job involves forecasting commodity prices. His explanation of his work is elaborate, and he stresses its complexity, and difficulty. Berry points out that carefully listening reduces the task to "smooth prices," which requires an APL function for exponential smoothing.

Some verbs are obvious; most PS will require retrieval and display routines such as "PLOT," "LIST," "FIND" or "TABLE." In some cases, the verb reflects a complex analytic routine, such as "PROJECT." This

routine invokes a dynamic programming model that involves complex assumptions and techniques. The key point made here is the direct relationships between managerial learning and extension of the PS, and between evolutionary use and evolutionary design. Obviously, the interface must be built so that new operators can be added easily and quickly.

A central difference between traditional approches to the design of computer tools and those needed for support systems is that the former assume completeness of specification and a "final" system. For PS, the issue is getting started, building a system that the user can relate to and that can encourage learning and evolution. The most effective approach for this seems to be to put most effort into the development of a "humanized" interface and a few key operators.

4.0 RESEARCH ISSUES RELATED TO PERSONAL SUPPORT

One test of the validity of the PS framework is whether a set of research issues can be formulated that will produce a deeper understanding of the issues. The research issues will be presented in two categories: (1) those issues that we need to know about before proceeding further with PS; and (2) gaps that become apparent in existing DSS research when viewed from a PS perspective.

4.1 IDENTIFYING IMPORTANT VERBS

For PS we need to know a great deal about the user and must develop effective methodologies for studying decision processes (Stabell terms this "Decision Research"). In particular, designers must be able to identify important verbs and define criteria for designing the software interface. In both cases the outputs from the process must be couched in a form that allows direct translation into software (and occasionally hardware) design. [Berry, 1977; Contreras, 1978] Some key research questions in this respect are:

1) How many important verbs do most decision makers use? Clearly, if the average is five, it will be far easier to get started with a PS than if it is 20 or 200.
2) What verbs do decision makers in different tasks have in common? Does there exist a set of primitive verbs that can form the initial Personal Support and be evolved into a useful Support System for a manager? It seems likely that there are some general verbs used for search and display of data ("FIND," "PLOT," "LIST," etc.).
3) What is a "good interface"? What are the users' measures of usability and quality? [Stirling, 1975; Little, 1975] At present "humanizing" or "customizing" the interface is a haphazard task with few validated rules.

These questions all focus on better ways of applying existing skills and technology for PS development. They are primarily empirical in nature. There are other more elusive conceptual issues to be resolved. Most obviously, if the distinction between PS and OS is valid, we need to revise the existing loose definitions of Decision Support. The cognitive focus of most DSS research [Keen & Scott Morton, 1978;

Stabell, 1974], seems fully applicable for PS, but there is relatively little work relevant to OS. The propositions of Galbraith [1973; 1977] about organizational information processing seem of practical value here.

4.2 LACK OF GOOD RESEARCH METHODOLOGY

Many of the gaps in our current knowledge are methodological and empirical. There is a complete absence of comparative research and effective use of case studies to validate or illustrate key concepts. Keen and Scott Morton [1978] provide some detailed descriptions of DSS. These suffer, however, from the lack of distinction between PS and OS; several of the systems are discussed as PS but are clearly OS (especially the Portfolio Management System which is the main example in the book). One conclusion to be drawn from the arguments presented here is that we need more descriptions of DSS usage; if the issue is getting started and then evolving the system, we must get a detailed understanding of how managers learn the general patterns of evolution. One key question is the extent and speed of change:

1) Do managers tend to rely on operators that support their existing verbs and not accept ones that require new concepts and modes of analysis?
2) What types of operators -- data based, model-based, heuristic, analytic, conceptual, etc. -- seem most beneficial or acceptable to the user? Are they based on individual differences, cognitive style, etc.?

Many of the questions above involve measurement. The tools for studying decision processes are primitive, at best. Protocol analysis, structured interviews and questionnaires are the most widely-used (and least validated) approaches. The problem is probably more complex for PS than OS. Organizational flows can be reliably identified [Hackathorn, 1977a], but cognitive processes are elliptic and often not observable. Stabell's argument, that the key issue for Decision Support is the development of methodologies for studying individual decision processes, seems of central relevance for PS. Regardless of available technologies, this rests on a comprehensive understanding of the individual user.

5.0 CONCLUSIONS

Most applications of Personal Computing have been simply extensions of programmable calculators, video games, and simple data retrieval. The PC journals provide few examples of practical uses that go much beyond checkbook balancing. At the same time the microprocessor-based hardware underlying PC has pushed towards small business systems with canned software packages for the typical clerical functions. The scope and direction of these developments have been determined largely by "technology push", rather than being pulled by the information processing needs of individuals.

PC requires a conceptual base for exploiting the potential of small-scale computer technology. Our argument here is that Personal Support provides such a framework. By definition, it involves discrete

tasks that do not require complex linkages with other systems and procedures. The aim of PC is essentially that of Decision Support: to provide individuals with tools under their own control so that they can adopt and adapt them to become extensions of their own problem solving processes.

Current concepts of PC focus very narrowly on the technology. As with Decision Support, the technology should be seen only as a means to an end. Unless PC defines that end. it has no systematic base for identifying applications and developing design strategies. The meshing of PS and PC constitutes that base. It seems clear that the technology of PC, software as much as hardware, can be used to provide cheap personal tools for managers. The cost factor is important mainly in reducing risk. That is, it is far easier to justify a Personal Support System whose initial hardware costs under $1.000 than one involving a larger investment and relatively high usage cost. Since PC relies on evolving larger systems from small prototypes, the tools of PC seem highly suited to the R and D process of building the initial version that is tailored to the needs and decision processes of a particular individual.

Personal Computing extends Decision Support. The challenge now is to carry out the decision research that can clarily where to apply the new technology. Just as Decision Support has moved from a somewhat vague rallying cry to an emergent discipline for research and practice, so too can Personal Computing build its distinctive technical base and provide decision makers with effective tools that they now lack.

REFERENCES

Alter, S. A. "A Study of Computer-aided Decision Making in Organizations." Ph.D. Dissertation, Sloan School, M.I.T., 1975.

Alter, S.A. "How Effective Managers Use Information Systems." Harvard Business Review, 1976.

Alter, S.A. Decision Support Systems: Current Practices and Future Challenges. Addison-Wesley, 1979. In Press.

Berry, P. "The Democratization of Computing." Paper presented at Eleventh Symposium Nacional de Systemas Computacionales, Monterrey, Mexico, March 15-18, 1977.

Carlisle, J. Cognitive Factors in Interactive Decision System. Ph.D. Dissertation, Yale University, 1974.

Carlson, E.D. (ed.) Proceedings of A Conference on Decision Support Systems. Database, vol. 8, no. 3, Winter 1977.

Contreras, L.E. Decision Support System and Corporate Planning. Informal Communication, 1978.

Galbraith, J.R. Designing Complex Organizations. Addison-Wesley, 1973.

Galbraith, J.R. Organizational Design. Addison-Wesley, 1977.

Gerrity, T.P. "The Design of Man-Machine Decision Systems: An Application to Portfolio Management." Sloan Management Review, vol. 12, no. 2, pp. 59-75. Winter 1971.

Gorry, G.A., and M.S. Scott Morton. "A Framework for Management Information Systems." Sloan Management Review, vol. 13, no. 1, pp. 55-77, Fall 1971.

Hackathorn, R.D. Modeling Unstructured Decision Making. Proceedings of the Conference on Decision Support Systems. Data Base, vol. 8, no. 3, pp. 41-42, Winter 1977 (a).

Hackathorn, R.D. "Design Criteria for Decision Support Systems." Proceedings of the 9th Annual Conference on the American Institute of Decision Science, Chicago. pp. 150-152, October 1977 (b).

Hackathorn, R.D. "Research Issues in Personal Computing." Proceedings of the National ACM Conference, Washington, December 1978. pp. 547-551.

Hoos, I.R. "Systems Analysis in Public Policy". Berkeley, University of California, 1972.

Isaacson, P. et. al. "The Oregon Report: Personal Computing." IEEE Computer, vol. 11, no. 9, pp. 86-97, September 1978.

Keen, P.G.W. "Computer Systems for Top Managers: A Modest Proposal". Sloan Management Review, Vol. 18. No. 1, pp. 1-17, 1976.

Keen, P.G.W. "The Intelligence Cycle: A Differentiated Perspective on Information Processing." American Federation of Information Processing Societies Proceedings, 1977 National Computer Conference, vol. 46, pp. 317-320, 1977.

Keen, P.G.W. and E.M. Gerson. "The Politics of Software Engineering." Datamation. pp. 80-86, Nov. 1977.

Keen, P.G.W. "Information Access For The Policy Analyst: A Portable Technology." Presented To the American Educational Finance Association, January 1979. To Appear in DS Working Paper Series.

Keen, P.G.W. "Decision Support Systems and the Marginal Economics of Effort." Working Paper No. 79-03-16. Department of Decision Sciences, The Wharton School, 1979.

Keen, P.G.W. and Hackathorn, R.D. "Studies of Decision Support Systems: A Review and Critique". Working Paper No. 79-03-17. Department of Decision Sciences, The Wharton School, 1979.

Kling, R. "The Organizational Context of User-centered Computer Program Designs." Working Paper, Department of Information and Computer Science, University of California, Irvine.

Licklider, J.C.R. Libraries of the Future. Cambridge: M.I.T. Press, 1965.

Little, J.D.C. "Brandaid." Operations Research, vol. 23, no. 4, pp. 628-673, May 1975.

Morton, M.S. Scott. Management Decision Systems: Computer Based Support for Decision Making. Cambridge, MA: Division of Research, 1971.

Ness, D.N. "Decision Support Systems: Theories of Design". Paper presented at Office of Naval Research Conference on Decision Support Systems, University of Pennsylvania, November 4-7 1975.

Ness, D.N. and Hurst, G.E. "Type P and Type S Systems." Working Paper 78-03-01. Department of Decision Science, The Wharton School, University of Pennsylvania, 1978.

Niles, J.M., Gray, P., Carlson, F.R, and Hayes, J. The Personal Computer and Society: A Technology Assessment. Presented at the National Computer Conference, Anaheim California, June 1978.

Stabell, C.B. On the Development of the Decision Support Systems as a Marketing Problem. Paper presented at the International Federation for Information Processing Congress, Stockholm, Sweden, August 1974.

Stabell, C.B. "Decision Research: Description and Diagnosis of Decision Making in Organizations." In D. Heradstreit and O. Narvesen (eds.), Decision Making Research: Some Developments. Oslo, Norway: Norsk Utenriks politisck Institute, 1977.

Sterling, T.D. "Humanizing Computerized Information Systems." Science, vol. 190, pp. 1168-1172, 19 December 1975.

Thompson, J.D. Organizations in Action. McGraw-Hill, 1967.

←1089 -79

1091-79

1093-79

1094-79

1095

1096-79

1097 -80

1098

1099-80

1100-80